Constructive Supervised Learning Algorithms for Artificial Neural Networks

Natalio Simon

Delft University of Technology
Faculty of Electrical Engineering

Type	**Master Thesis**
Number of Pages	97
Date	4 juni 1993
Department	**Computer Architecture and Digital Technique**
Codenumber	**1-68340-28(1993)06**
Author	**Natalio Simon**
Title	**Constructive Supervised Learning Algorithms for Artificial Neural Networks**
Supervisor	Prof.dr.ir. A.J. van de Goor
Mentors	Dr. J. Hoekstra

Abstract

The field of Artificial Neural Networks has received in the last years a great deal of attention from the academic community and the industry. Neural Network applications are now found in a wide range of areas, including Control Systems, Medicine, Communication, Cognitive Sciences, Linguistic, Robotics, Physics and Economics.

This report focuses on one of the key challenges of the field of Artificial Neural Networks, the creation of good general-purpose learning algorithms.

As opposed to conventional programmed computing, where a series of explicit commands (a program) processes information on an associated database, Artificial Neural Networks develop information processing capabilities by learning from examples.

Learning techniques can be roughly divided into two categories: Supervised Learning, and Self-Organization or Unsupervised Learning.

Supervised learning requires a set of examples for which the desired network response is known. The learning process consists then of adapting the network in a way that it will produce the correct response for the set of examples. The resulting network should then be able to generalize (give a good response) when presented with cases not found in the set of examples.

The class of Constructive Supervised Learning Algorithms has been defined in this report. As opposed to non-constructive learning algorithms, where the topology of the network is selected before the algorithm starts and remains fixed, Constructive Supervised Learning Algorithms start from an initial network topology and they add new elements (neurons) in order to learn the set of examples. The final topology and the size of the network are then dynamically determined by the algorithm and are a function of the set of examples chosen and of the learning parameters.

A division has been made between Constructive Supervised Learning Algorithms for continuous connectionist networks, and Constructive Supervised Learning Algorithms for discrete connectionist networks.

The power and applicability of Constructive Supervised Learning Algorithms has been shown by using them in six applications of different nature. The applications are:

1. Solution to the Inverse Kinematic Transformations for Robot Control.
2. Corrective networks for damage Robot Controllers.
3. Hamming Codes Generators and Parity Generators.
4. A Pattern Recognition problem.
5. Mapping of Random Boolean Functions.
6. The Monk's problems.

The above applications have allowed an analysis and comparison of the different constructive learning algorithms presented. The comparison has been done in terms of learning speed, size of the network and generalization accuracy.

Continuous Constructive Supervised Learning Algorithms

For Continuous Constructive Supervised Learning Algorithms, the Cascade-Correlation Learning Algorithm and a modified version of it have been presented.

The Cascade-Correlation Algorithm builds a network by adding a hidden unit at a time between the previously added hidden units and the output units, until the training set is learned. Hidden units are trained to correlate to the sum of the remaining outputs' errors before they are connected.

The Modified Cascade-Correlation Algorithm has been introduced for problems were multiple outputs are required. In this algorithm, a new unit is added per output and is trained to correlate to the remaining error at the respective output before it is connected. Then the hidden unit is connected to all output units.

The Inverse Kinematic Transformations for a two and a three degrees of freedom Robot Controller has allowed for a comparison between the Modified and the original Cascade-Correlation algorithms, For this problem, the Modified Cascade-Correlation was shown to learn quicker, scale better to more training cases and to lower tolerance, and to have a higher success rate than the original Cascade-Correlation.

The Inverse Kinematic Transformation for a two degrees of freedom Robot Controller allowed also a comparison of the Modified Cascade-Correlation algorithm with previous results obtained with the non-constructive Backpropagation learning algorithm. In this case the accuracy of generalization achieved by the Modified Cascade-Correlation algorithm was equal or better and with a much smaller network than achieved by using the Backpropagation algorithm.

The other advantage given by the use of the Modified Cascade-Correlation algorithm over the Backpropagation algorithm is that no trial and error stage is needed to determine the amount and size of the hidden layers in the network, what is the case with the non-constructive Backpropagation algorithms.

The determination of an a-priori network topology when applying non-constructive learning algorithms, so as the Backpropagation algorithm, remains a difficult problem in the area of Robot Control. The results achieved for the Inverse Kinematic Transformation and for a Corrective Networks for a damage Robot Controllers, point to the applicability of the Modified Cascade-Correlation learning algorithm for this area of application.

For the Pattern Recognition problem and for the Hamming Code Generation problem, the Modified Cascade-Correlation Learning Algorithm was also shown to learn faster than the original Cascade-Correlation Learning Algorithm, When tested for generalization accuracy with the Inverse Kinematic Transformations and the Corrective networks for a damage Robot Controllers, the Modified Cascade-Correlation algorithm also scored better.

Discrete Constructive Supervised Learning Algorithms

For Discrete Constructive Supervised Learning Algorithms, the Tilling Learning Algorithm, the Tower Algorithm, the Upstart Algorithm and a new algorithm introduced in this report, the BINCOR Algorithm, have been presented.

The Tilling Learning Algorithm and the Tower Learning Algorithm build a network by adding a layer at a time, where the internal representation of the input patterns in each layer remains different for input patterns with different outputs. The process goes on until the first unit added at a new layer produces the correct mapping for all input patterns. The convergence of these algorithms for any Boolean mapping is guaranteed. The Tilling Algorithm adds a variable number of units in each layer. The Tower Algorithm adds only one unit in each layer and uses "shortcut" connections, i.e. it not only connects the unit of a new layer to the unit of the previous layer, so as in the Tilling Algorithm, but also to the input units.

The Upstart Algorithm and the BINCOR Algorithm use a different strategy than the previous two algorithms for building a network. In the Tilling and the BINCOR algorithms, units are added between the output unit and the inputs in order to correct the remaining misclassified patterns at the output unit. The convergence of these algorithms for any Boolean mapping is guaranteed. In the Upstart Algorithm a recursive strategy is used, where each unit added receives connections from the input units and tries to correct the misclassified input patterns at the unit previously added, leading to a binary tree network structure. In the BINCOR Algorithm, two units are added at a time between the input units and the output unit, where this units try to correct the remaining misclassified input patterns at the output unit, leading to one of two optional network structures: a single hidden layer or a high-order network of multiple layer, where the units in a hidden layer receive connections from the input units and all previously added units.

The mapping of random Boolean functions has allowed for a comparison between the Tilling, Upstart and BINCOR algorithms in terms of size of the network built. The introduced BINCOR algorithm produced the network with less units of the three algorithms.

The Monk's problems has been used to compare the generalization accuracy of the BINCOR algorithm with a variety of machine learning algorithms. The generalization with the BINCOR algorithm was less accurate than with the Cascade-Correlation Learning Algorithm and the Backpropagation Learning Algorithm, but more accurate than with Decision Tree-Based Learning Algorithms, the mFOIL algorithm and Inductive Learning programs.

Contents

Introduction

1

After having being forgotten for more than two decades, the field of Artificial Neural Networks has received in the last years a great deal of attention from the academic community and the industry.

Characteristic to the field is its multidisciplinary character. Neural Network articles are periodically found in Control Systems, Medicine, Communication, Cognitive Sciences, Linguistic, Robotics, Physics, Economics and Artificial Intelligence journals. Added to this are the journals specific to the field of Neural Networks (see Table 1.1).

Neural Networks have also seen their way to industry in the last two years. Table 1.2 presents a list of the major holders of U.S. Neural Network patents through 1991 [Wenskay, 1992]. This list includes leading companies in Electronics so as Philips, IBM, Intel, Mitsushita and Kodak.

Governments are also encouraging developments in this field. The European Community, as an example, funds several R & D projects in Neural Networks under its ESPRIT II and ESPRIT III programmes [Corsi, 1992], [Angeniol et al, 1992] (see Table 1.3). These are being carried out jointly by industry and research centers.

This chapter begins with an overview of Artificial Neural Networks and a survey of the techniques for learning in Artificial Neural Networks. Then the class of constructive learning algorithms is presented. Finally, a brief overview of the structure of the rest of this report is given.

1.1 What are Artificial Neural Networks

Artificial Neural Networks are the primary information processing structures of a technological discipline called Neurocomputing. Neurocomputing is concerned with parallel, distributed, adaptive information processing systems that develop information processing capabilities in response to exposure to an information environment [Hecht-Nielsen, 1990]. Artificial Neural Networks are applied typically in areas such as sensor processing, patterns recognition, data analysis and control, which may require information processing structures for which the algorithms or rules are not known.

An Artificial Neural Network is in the most general sense an information processing structure whose design is motivated by the design and functioning of human brains and components thereof. The formal definition of a neural network follows.

Definition A neural network is a parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches ("fans out") into as many collateral connections as desired; each carries the same signal - the processing element output signal. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via incoming connections and on values stored in the processing element's local memory.

A neural network can be categorized into:

1. **Continuous Neural Network**, where the input and output space are of a continuous nature, and
2. **Discrete Neural Network**, where the input and output space are of a discrete nature.

As opposed to conventional programmed computing, where a series of explicit commands (a program) processes information on an associated database, Artificial Neural Networks develop information processing capabilities by learning from examples.

Learning techniques can be roughly divided into two categories:

1. Supervised Learning.
2. Self-organization or Unsupervised Learning

Supervised learning requires a set of examples for which the desired network response is known. The learning process consists then in adapting the network in a way that it will produce the correct response for the set of examples. The resulting network should then be able to generalize (give a good response) when presented with cases not found in the set of examples.

In self-organization or unsupervised learning the neural network is autonomous: it processes the data it is presented with, finds out about some of the properties of the data set and learns to reflect these properties in its output. What exactly these properties are, that the network can learn to recognize, depends on the particular network model and learning method.

Many of the learning techniques are closely connected with a certain (class of) network topology.

1.2 Current state of Artificial Neural Networks

Attention in Artificial Neural Networks is being focused primarily in four areas:

1. Software simulation.
2. Theoretical foundations.
3. Hardware implementation.
4. Applications.

Software simulation

Most studies carried out to compare the performance of the different learning algorithms and network topologies are being done on serial computers that simulate the the operation and flow of data of the different models of neural networks.

For this purpose, various benchmark problems are used. Among the most common [Fahlman, 1988], [Fahlman & Lebiere, 1990]:

- The XOR/Parity Problem.
- The Encoder/Decoder Problem.
- The Two-Spirals Problem.

The restricted scope of these benchmarks has led to the use of real-world problems, where higher number of inputs and outputs are used, to compare the performance of the different algorithms and topologies.

Commonly used real-world problems are:

- Handwritten character set recognition.
- Inverse Kinematic Transformations for Robot Control.
- Time-Series prediction.

The various algorithms are compare in terms of:

- Learning speed.

- Size of the network.
- Generalization accuracy.

By *learning speed*, the number of connection crossing or the number of epochs (one epoch is one cycle over the training set) are used as a measure.

The *size of the network* is given by the amount of neurons and the number of layers (and the number of neurons per layer).

By *generalization accuracy* it is meant how well the network performs for input patterns not found in the training set. For this measurement a *test set* is used and the average, maximum and standard deviation of the network error for the test set is compared.

For real-world problems, the training set represents a small part of the input space so the way this set is chosen to important to how well the network learns. Good learning is reflected in good generalization accuracy.

Another factor that plays a roll in how well the networks learns is the error tolerance in learning the training set. If the training set is too accurately learned, then the *over-training* or *grand-mothering* problem occurs. In this case the network learn the training set very well but generalizes poorly for input patterns outside the training set.

Theoretical foundations

Neural Networks are able to learn from examples, to approximate complex functions, to make predictions of non-linear system outputs. They are used for the diagnosis of faulty equipment, to determine the trajectory of a robot, etc. Still a complete formal foundation of how the various algorithms "learn" and how to achieve optimal learning and optimal network size is lacking. Research is carried out to give a mathematical foundation to the different learning algorithms and network topologies.

Hardware implementation

Artificial Neural Networks are inherently parallel. They consist of simple processing units (neurons) that process data independently of each other. Simulations on conventional sequential computers provide a functionality test. To fully exploit the high-speed potential of neural networks for problems requiring real-time response, work is done in the implementation of neural networks in physical hardware, where each processing element can operate independent and in parallel to other elements. Approaches in hardware implementation include analog, digital as well as hybrid circuits.

Two options exist regarding the implementation of Artificial Neural Networks in hardware: implementing hardware for a specific application and for a specific learning algorithm and topology, or a general purpose architecture suitable for different learning algorithms and topologies and for applications with different sizes of inputs and outputs. The last approach being followed by many researchers and the industry (see [Barhen et al., 1992],[Daud et al., 1992],[Hurdle et al., 1992], [Jutten et al., 1990], [Corsi, 1992]).

Applications

Neural Networks are in a stage of development were researchers continue to discover new applications for this technology. From 1989 there is being an exponential increase in patent activity in Neural Networks, lead by the U.S. with Japan in the second place, followed by Korea, Germany, France and the United Kingdom [Wenskay, 1992].

Neural Networks are applied in a wide range of tasks. These tasks include, among others,

- Quality Control: of video images, VLSI masks, leather, oranges.
- Forecasting: of electricity consumption, economic and financial time-series, chip dimensions.
- Vision: motion detection, recognition of objects, texture.
- Diagnosis: of metal corrosion, machinery.
- Control: robot control, industrial process control.
- Classification: of patterns, spectra.

The impact of Artificial Neural Networks is likely to be in areas of artificial perception - making sense of speech, language and visual images. This is where the promise of going beyond the capabilities of conventional programming arise, simply because conventional programming requires an algorithm which, in turn, requires a full analysis of the problem. But because the mechanisms used by humans when they carry out these tasks are not fully understood, satisfactory algorithms cannot be devised. The expectation is that Artificial Neural Networks will build up such solutions through being trained on appropriate input-output relationships [Aleksander & Morton].

1.3 Constructive Learning Algorithms

This report focuses on a class of *Supervised Learning Algorithms* for Artificial Neural Networks, called Constructive or Dynamic Learning Algorithms.

Constructive or Dynamic Learning Algorithms have the characteristic that they modify the neural network topology in the process of learning from a set of examples. They start from an initial network topology and they add new elements (neurons) in order to learn the set of examples. The final topology and the size of the network are dynamically determined by the algorithm and are a function of the set of examples chosen and of the learning parameters.

The aim of this report is threefold:

1. to present the class of Constructive or Dynamic Learning Algorithms,
2. to show the power and applicability of these algorithms by using them in applications of different nature, and
3. to analyze their advantages and disadvantages in comparison with other non-constructive learning algorithms.

Title:	Neural Networks	Title:	Neural Computation
Publisher:	Pergamon Press	Publisher:	MIT Press
Year:	volume 1 in 1988	Year:	volume 1 in 1989
Title:	Neurocomputing	Title:	IEEE Transaction on Neural Networks
Publisher:	Elsevier	Publisher:	IEEE
Year:	volume 1 in 1989	Year:	volume 1 in 1990
Title:	International Journal of Neural Systems	Title:	Network: Computation in Neural Systems
Publisher:	World Scientific Publishing	Publisher:	IOP Publishing Ltd
Year:	volume 1 in 1990	Year:	volume 1 1990
Title:	Journal of Neural Network Computing, Technology, Design, and Applications	Title:	Connection Science: Journal of Neural Computing Artificial Intelligence and Cognitive Research
Publisher:	Auerback Publishers	Publisher:	Carfax Publishin
Year:	volume 1 1989	Year:	volume 1 in 1989
Title:	International Journal of Neural Networks	Title:	Neural Computing and Applications
Publisher:	Learned Information	Publisher:	Springer International
Year:	volume 1 in 1989	Year:	volume 1 in 1992
Title:	Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies		
Publisher:	Kluwer Academic Publishers		
Year:	volume 1 in 1991		

Table 1.1: Neural Network Journals

1.4 Organization of the Report

The report is organized as follows:

Chapter 2 presents the class of non-constructive supervised learning algorithms. Here, for discrete networks, the Perceptron Learning Algorithm is discussed. Then, for continuous networks, the Backpropagation Algorithm and two variations of it, the Quickprop Algorithm and the Bold-Drive Method, are discussed.

Chapter 3 presents the class of constructive supervised learning algorithms for continuous networks. Here the Cascade Correlation Algorithm and variations on it are discussed. Then the performances of these and algorithms presented in chapter 2 are compared. The comparison is done for the following applications: a pattern recognition problem, the Hamming Code generation problem, and the inverse kinematic transformations of a two and a three degrees of freedom robot arm.

Chapter 4 presents the class of constructive supervised learning algorithms for discrete networks. Here the Pocket Learning Algorithm, the Tilling Algorithm and the BINCOR Algorithm are discussed. Then the performances of these and algorithms presented in the previous chapters are compared. The comparison is done for the following applications: the Monk's Problem and random boolean functions.

Chapter 5 finalizes this report with a discussion of the performances obtained by the learning algorithms of the previous chapters. Conclusions regarding the suitability of Constructive Learning Algorithms are given. An outline of future work directions is presented.

Miscellaneous Univ.	AT&T
IBM	U.S. Government
Hughes Aircraft	Nestor
GTE	Synaptics
Matsushita	Intel
General Electrics	Philips
Kodak	Motorola
Hitachi	

Table 1.2: Major Holders of U.S. Neural Network Patents Though 1991

UNDER ESPRIT II 1988-1990	UNDER ESPRIT III 1990-
PIGMALION *Neurocomputing*	CONNY *Robot Control Based on Neural Network Systems*
GALATEA *Neurocomputing*	HIMMARNET *Study of Hidden Markov Models and Neural Networks for Robust Isolated Word Recognition*
ANNIE *Applications of Neural Networks for Industry in Europe*	NEUROQUACS *Neural Networks based Vision and Signal System for Industrial Quality Control*
NEUFODI *Neural Networks for Forecasting and Diagnosis Applications*	WERNICKE *Neural Network Based, Speaker Independent, Large Vocabulary, Continuous Speech Recognition System*
SPRINT *Speech Processing and Recognition Using Integrated Neurocomputing Techniques*	MUCOM *Multisensory Control of Movement*
SUBSYM *Self-Organization and Analogical Modeling Using Sub-Symbolic Computing*	INSIGHT *Vision Systems for a Natural Human Environment*
NERVES *Innovative Architectures for Neurocomputing Machines and VLSI Neural Networks*	ELENA-NERVES 2 *Enhanced Learning for Evolutive Neural Architectures*
	SSS *Smart Sensory Systems*

Table 1.3: EC Neural Network Projects

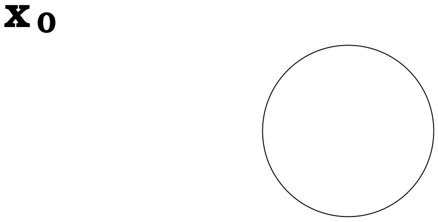
Non-Constructive Supervised Learning

2

Supervised learning consists of training a neural network with a set of examples for which the desired outputs are known. The purpose of the learning algorithm is then to adjust the network so that the network produces the correct outputs for the set of examples.

Here we distinguish between two classes of Learning Algorithms: *Non-Constructive* and *Constructive*. By non-constructive we refer to algorithms for which the topology of the network has to be fixed a-priori, while in constructive ones the algorithm itself automatically determines the topology of the network.

This chapter presents the class of non-constructive supervised learning algorithms. First the Perceptron Learning Algorithm [Rosenblatt, 1962] is described and discussed. Then the Backpropagation Algorithm [Rumelhart, Hinton & Williams, 1986] and two variations of it, the Quickprop Algorithm [Fahlman, 1988] and the Bold-Drive Method [Vogl et al.], are described and discussed.


$\mathbf{x}_0$